WRITTEN REPORT OF THE INTERNATIONAL SEARCH AUTHORITY

Supplementary Page T/EP2004/050465

Regarding Item V:

1 Reference is made in this Report to the following documents:

D1: WO 01/46806 A (INTEL CORP; QUACH NHON TOAI (US)) June 28, 2001

D2: US-A-5 043 990 (DOI TOSHIO ET AL) August 27, 1991

D3: US-A-5 640 508 (FUJIWARA HIROKATSU ET AL) June 17, 1997

D4: HENNESSY, PATTERSON: "Computer Architecture" June 30, 2002, MORGAN KAUFMANN, XP002305487

2 DEPENDENT CLAIM 1

2.1 The present application does not meet the requirements of PCT Article 33(1) because the subject matter of Claim 1 is not based on an inventive step as defined in PCT Article 33(3).

2.1.1 Document D1 is regarded as the closest prior art with respect to the subject matter of Claim 1. It discloses (citations in parentheses refer to D1; the original wording of the claim is in italics; struck-through passages are not disclosed in D1; underlined passages are additionally disclosed in D1) a:

Program-controlled unit ~~comprising a single controller core that~~ which has a first and at least a second execution unit (Abstract, "processor having dual execution cores"), which units are operable independently of one another in a first operating mode (page 2, "SMP mode"), and process the same

EV 331378175 US

instructions in parallel in a second operating mode (page 2, "FRC mode").

2.1.2 The subject matter of Claim 1 therefore differs from what is known from D1 in that the program-controlled unit possesses only a single controller core having two redundant arithmetic units. The program-controlled unit of D1, on the other hand, possesses two redundant controller cores. The technical effect that results from this difference is that the program-controlled unit according to Claim 1 requires less chip area.

2.1.3 The object to be achieved with the present invention can thus be seen as that of reducing the chip area requirement of a redundant program-controlled unit, while retaining redundancy.

2.1.4 The manner of achieving this object proposed in Claim 1 of the present Application cannot be regarded as inventive for the following reasons (PCT Art. 33(3)):

The feature of equipping a controller core with a redundant arithmetic unit has, however, already been used for the same purpose in a similar program-controlled unit; see document D2, especially col. 1, lines 22-22 (sic) and Fig. 1. If one skilled in the art wishes to achieve the same purpose in a program-controlled unit according to document D1, it is entirely possible for him to apply the features, with corresponding effect, in the context of the subject matter of document D1 as well. He would in that fashion arrive, without inventive merit, at a program-controlled unit according to Claim 1.

2.1.5 The manner of achieving the object proposed in independent Claim 1 therefore cannot be regarded as inventive (PCT Article 33(3)).

3. INDEPENDENT CLAIM 11

3.1 The reasoning of section 2 applies correspondingly to independent Claim 11.

3.2 The additional feature of comparing the calculated result data with one another, and generating an error signal in the event of non-agreement, is also known from D1 (col. 1, lines 25-29).

3.3 The subject matter of Claim 11 is therefore not based on an inventive step (PCT Article 33(3)).

4. DEPENDENT CLAIMS 2-10, 12-15

Claims 2 through 10 and 12 through 15 contain no features that, in combination with the features of any claim to which they refer, meet the requirements of the PCT with regard to novelty or inventive step.

4.1 With reference to Claim 2, document D2 (Figs. 1 and 2, "comparison unit") also discloses an error detection device.

4.2 With reference to Claim 3, document D2 (Figs. 1 and 2, "parity generation") also discloses a coder.

4.3 With reference to Claim 4, document D2 (Figs. 1 and 2, "comparison unit") also discloses a comparison unit downstream from the two execution units on the output side.

4.4 With reference to Claim 5, document D2 (Figs. 1 and 2, "parity check") also discloses a comparison unit upstream from each of the execution units on the input side.

4.5 With reference to Claim 6, document D2 (Fig. 2, "register 1501," "register 1502") also discloses data registers that are associated with the execution units.

4.6 With reference to Claims 7-9 and 14, shadow registers for fallback to older operand values in the event of error are known to one skilled in the art; see e.g. D4 (page A-55, "history file").

4.7 With reference to Claim 10, the construction of the program-controlled unit as a microcontroller or microprocessor is obvious.

4.8 With reference to Claim 5, document D2 (Fig. 2, "E, E_A, E_B") also discloses an error signal for each type of error.

4.9 With reference to Claim 13, it is disclosed in document D2 (Fig. 2, "E_A, E_B," "F") that the input data are first conveyed to both execution units and subsequently thereto the error correction code is created from the input data.

4.10 With reference to Claim 15, it is obvious to one skilled in the art that the result data are placed onto the bus only if an error signal is not present.

Regarding item VIII:

Specific comments on the International Application

5. The Application does not meet the requirements of PCT Article 6 because the following claims are not clear:

5.1 A lack of clarity may result from the expressions "core" (omitted from English translation) and "e.g. parity, CRC, ECC" placed in parentheses in Claims 1 and 5.

5.2 Claims 3 and 11 are not clear because they contain a plurality of alternatives ("and/or") that result in difficulties in construction.

5.3 Claim 11 can be construed in such a way that the input data of the two calculation units are compared with one another. This is not supported, however, by the specification or the drawings.

(handwritten notes)

R 304832 Mr. Kolke: (?sp) February 4, 2005

Different logic (not ALU) can easily be protected e.g. by parity.

With the ALU, however, single errors keep slipping through:

=> Create redundancy with regard to ALUs

=> Double ALU can be better protected using error codes (such as ECC). Therefore double-ALU configuration also means protection of the transition of this different logic to double ALU, therefore continuous protection against errors despite different error discovery method (different logic ≈ parity & (?) double ALU ≈ redundancy)! Normally there are gaps in protection at X!